Project

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from google.colab import drive

drive.mount('/content/drive')

path='/content/drive/MyDrive/clean data3.xlsx'

df=pd.read_excel(path)

print(df)

```
    Company   TypeName   Ram  Weight  Price_euros  TouchScreen  Ips       Ppi  \
0     Apple  Ultrabook   8GB  1.37kg      1339.69            0    1  226.9830   
1     Apple  Ultrabook   8GB  1.34kg       898.94            0    0  127.6779   
2        HP   Notebook   8GB  1.86kg       575.00            0    0  141.2120   
3     Apple  Ultrabook  16GB  1.83kg      2537.45            0    1  220.5346   
4     Apple  Ultrabook   8GB  1.37kg      1803.60            0    1  226.9830   
..      ...        ...   ...     ...          ...          ...  ...       ...   
995    Asus   Notebook   8GB   1.4kg      1150.00            0    0  276.0535   
996      HP  Ultrabook   8GB  1.11kg      1349.00            1    1  165.6321   
997    Acer   Notebook   4GB   2.4kg       380.00            0    0  100.4547   
998    Asus     Gaming  16GB   2.5kg      1799.00            0    0  141.2120   
999      HP  Ultrabook   8GB  1.48kg      2089.00            0    0  157.3505   

                 Cpu_brand   HDD  SSD       Os  
0            Intel Core i5     0  128      Mac  
1            Intel Core i5     0    0      Mac  
2            Intel Core i5     0  256   Others  
3            Intel Core i7     0  512      Mac  
4            Intel Core i5     0  256      Mac  
..                     ...   ...  ...      ...  
995          Intel Core i5     0  256  Windows  
996          Intel Core i7     0  256  Windows  
997  Other Intel Processor   500    0  Windows  
998          Intel Core i7  1000  256  Windows  
999          Intel Core i7     0  512  Windows  

[1000 rows x 12 columns]
```

null_values=df.isnull().sum()

print('null values in dataframe')

print(null_values)

```
null values in dataframe
Company        0
TypeName       0
Ram            0
Weight         0
Price_euros    0
TouchScreen    0
Ips            0
Ppi            0
Cpu_brand      0
HDD            0
SSD            0
Os             0
dtype: int64
```

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Company      1000 non-null   object
 1   TypeName     1000 non-null   object
 2   Ram          1000 non-null   object
 3   Weight       1000 non-null   object
 4   Price_euros  1000 non-null   float64
 5   TouchScreen  1000 non-null   int64
 6   Ips          1000 non-null   int64
 7   Ppi          1000 non-null   float64
 8   Cpu_brand    1000 non-null   object
 9   HDD          1000 non-null   int64
 10  SSD          1000 non-null   int64
 11  Os           1000 non-null   object
dtypes: float64(2), int64(4), object(6)
memory usage: 93.9+ KB
```

df.describe()

	Price_euros	TouchScreen	Ips	Ppi	HDD	SSD
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	1125.706070	0.145000	0.283000	147.261968	418.660000	190.864000
std	705.380659	0.352277	0.450682	42.531903	531.764973	188.651284
min	191.900000	0.000000	0.000000	90.583400	0.000000	0.000000
25%	599.000000	0.000000	0.000000	127.335700	0.000000	0.000000
50%	959.500000	0.000000	0.000000	141.212000	0.000000	256.000000
75%	1474.250000	0.000000	1.000000	157.350500	1000.000000	256.000000
max	6099.000000	1.000000	1.000000	352.465100	2000.000000	1024.000000

df['Company'].value_counts()

```
Dell         237
Lenovo       230
HP           185
Asus         126
Acer          84
Toshiba       41
MSI           34
Apple         17
Samsung        8
Mediacom       7
Microsoft      6
Razer          5
Xiaomi         4
Chuwi          3
Google         3
Fujitsu        3
LG             3
Huawei         2
Vero           2
Name: Company, dtype: int64
```

df['TypeName'].value_counts()

```
Notebook              563
Ultrabook             154
Gaming                153
2 in 1 Convertible     88
Workstation            24
Netbook                18
Name: TypeName, dtype: int64
```

df['Cpu_brand'].value_counts()

```
Intel Core i7            395
Intel Core i5            339
Other Intel Processor    111
Intel Core i3            106
AMD Processor             49
Name: Cpu_brand, dtype: int64
```

df['Os'].value_counts()

```
Windows    859
Others     124
Mac         17
Name: Os, dtype: int64
```

df.groupby('Company').mean()

Company	Price_euros	TouchScreen	Ips	Ppi	HDD	SSD
Acer	628.037024	0.107143	0.380952	128.452739	398.809524	93.142857
Apple	1655.715882	0.000000	0.764706	201.999676	0.000000	233.411765
Asus	1089.003254	0.111111	0.134921	139.127092	580.380952	182.349206
Chuwi	314.296667	0.000000	0.333333	183.254133	0.000000	0.000000
Dell	1238.045148	0.189873	0.160338	152.653132	502.109705	201.485232
Fujitsu	729.000000	0.000000	0.000000	100.454700	333.333333	170.666667
Google	1677.666667	1.000000	0.000000	234.507400	0.000000	298.666667
HP	1028.066811	0.124324	0.248649	143.891410	381.081081	171.459459
Huawei	1424.000000	0.000000	1.000000	199.692100	0.000000	384.000000
LG	2099.000000	0.666667	1.000000	146.591500	0.000000	512.000000
Lenovo	1078.845174	0.139130	0.439130	148.335169	356.521739	209.147826
MSI	1789.748235	0.000000	0.117647	142.014947	1000.000000	256.000000
Mediacom	295.000000	0.142857	0.714286	164.992414	4.571429	18.285714
Microsoft	1612.308333	1.000000	0.000000	200.842600	0.000000	256.000000
Razer	3779.000000	0.600000	0.200000	235.301740	0.000000	558.400000
Samsung	1507.750000	0.250000	0.000000	151.723000	125.000000	208.000000
Toshiba	1219.365854	0.121951	0.365854	138.808241	109.756098	218.536585
Vero	231.450000	0.000000	0.500000	157.350500	0.000000	0.000000
Xiaomi	1133.462500	0.000000	1.000000	153.422050	0.000000	256.000000

df['TypeName'].unique()

print(df['TypeName'].unique())

df['TypeName'].nunique()

```
['Ultrabook' 'Notebook' 'Netbook' 'Gaming' '2 in 1 Convertible'
 'Workstation']
6
```

df.corr()

	Price_euros	TouchScreen	Ips	Ppi	HDD	SSD
Price_euros	1.000000	0.206781	0.233805	0.498427	-0.085800	0.672259
TouchScreen	0.206781	1.000000	0.100658	0.438605	-0.185452	0.244080
Ips	0.233805	0.100658	1.000000	0.303626	-0.108383	0.214483
Ppi	0.498427	0.438605	0.303626	1.000000	-0.275068	0.524033
HDD	-0.085800	-0.185452	-0.108383	-0.275068	1.000000	-0.391658
SSD	0.672259	0.244080	0.214483	0.524033	-0.391658	1.000000

```
pd.pivot_table(df,values='Price_euros',index='Company',columns='TypeName',aggfunc='mean')
```

TypeName	2 in 1 Convertible	Gaming	Netbook	Notebook	Ultrabook	Workstation
Company						
Acer	634.430000	1314.333333	339.000000	563.652381	890.500000	NaN
Apple	NaN	NaN	NaN	NaN	1655.715882	NaN
Asus	976.385385	1672.670000	258.966667	638.887037	1396.112500	NaN
Chuwi	NaN	NaN	NaN	314.296667	NaN	NaN
Dell	1114.742632	1993.353333	519.500000	915.349091	1464.862973	2219.169091
Fujitsu	NaN	NaN	NaN	729.000000	NaN	NaN
Google	NaN	NaN	NaN	NaN	1677.666667	NaN
HP	1447.524375	1506.100000	1366.800000	757.293689	1435.782174	2191.572222
Huawei	NaN	NaN	NaN	NaN	1424.000000	NaN
LG	NaN	NaN	NaN	NaN	2099.000000	NaN
Lenovo	1638.174839	1201.478261	431.000000	776.216503	1842.346538	2381.000000
MSI	NaN	1789.748235	NaN	NaN	NaN	NaN
Mediacom	299.000000	NaN	NaN	294.333333	NaN	NaN
Microsoft	NaN	NaN	NaN	NaN	1612.308333	NaN
Razer	NaN	4274.000000	NaN	NaN	1799.000000	NaN
Samsung	1799.000000	NaN	269.000000	1699.000000	1659.000000	NaN
Toshiba	NaN	NaN	NaN	1081.375000	1710.000000	NaN
Vero	NaN	NaN	NaN	231.450000	NaN	NaN
Xiaomi	NaN	NaN	NaN	1299.475000	967.450000	NaN

```
df.sort_values('Price_euros',ascending=False)
```

	Company	TypeName	Ram	Weight	Price_euros	TouchScreen	Ips	Ppi	Cpu_brand	HDD	SSD	Os
196	Razer	Gaming	32GB	3.49kg	6099.0	1	0	254.6713	Intel Core i7	0	1000	Windows
830	Razer	Gaming	32GB	3.49kg	5499.0	1	0	254.6713	Intel Core i7	0	512	Windows
610	Lenovo	Notebook	32GB	2.5kg	4899.0	0	1	282.4240	Other Intel Processor	0	1000	Windows
749	HP	Workstation	16GB	3kg	4389.0	0	1	127.3357	Other Intel Processor	0	256	Windows
238	Asus	Gaming	32GB	4.7kg	3890.0	0	0	127.3357	Intel Core i7	1000	512	Windows
...	...	...	...	...	...	...	...	...	...	...	...	...
555	Asus	Notebook	4GB	2kg	224.0	0	0	100.4547	Other Intel Processor	500	0	Others
791	Vero	Notebook	4GB	1.22kg	202.9	0	0	157.3505	Other Intel Processor	0	0	Windows
31	Asus	Notebook	2GB	1.65kg	199.0	0	0	111.9352	AMD Processor	0	0	Windows
290	Acer	Notebook	2GB	2.19kg	199.0	0	0	100.4547	Other Intel Processor	0	16	Others
20	Asus	Netbook	2GB	0.98kg	191.9	0	0	135.0942	Other Intel Processor	0	0	Windows

1000 rows × 12 columns

df['Price_euros'].min()

```
Price_euros    191.9
dtype: float64
```

df[['Price_euros']].max()

```
Price_euros    6099.0
dtype: float64
```

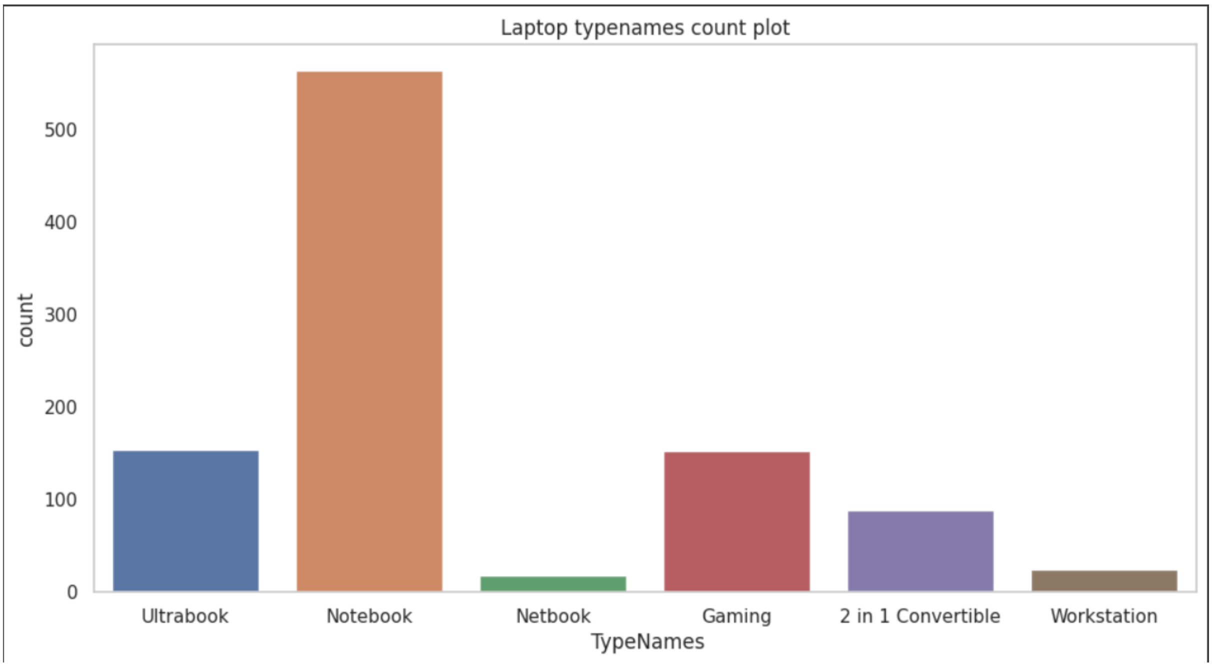
Univariate Analysis

```
price=df['Price_euros']


mean = price.mean()
median = price.median()
mode = price.mode()[0]
std_dev = price.std()
print(f"Mean: {mean}")
print(f"Median: {median}")
print(f"Mode: {mode}")
print(f"Standard Deviation: {std_dev}")
```

```
Mean: 1125.7060699999997
Median: 959.5
Mode: 1099.0
Standard Deviation: 705.3806589625871
```

What is the total count of types of laptop?

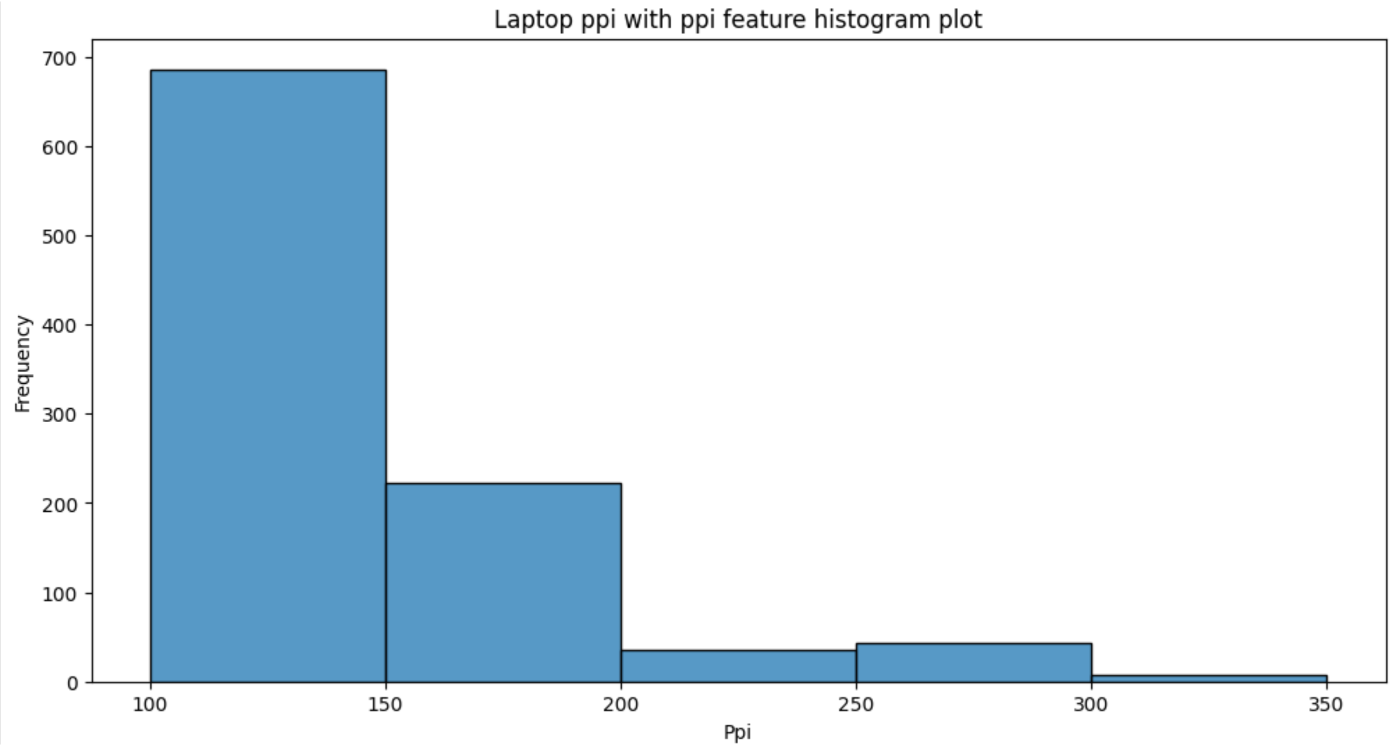
```
plt.figure(figsize=(17,6))

sns.countplot(x=df['TypeName'])

plt.title("Laptop typenames count plot")

plt.xlabel(" TypeNames")

plt.ylabel("count")

plt.grid(False)
```



Above graph shows count of types of laptop. From above graph we conclude that Notebook type of laptop is maxmimum in number and Netbook type of laptop is least in number

What is the frequency of laptop price?

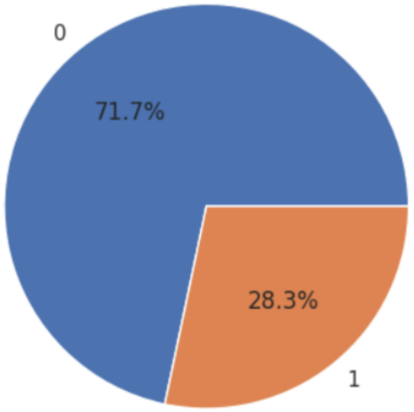
```
plt.figure(figsize=(12,6))
sns.histplot(x=df['Ppi'],bins=[100,150,200,250,300,350])
plt.title("Laptop ppi with ppi feature histogram plot")
plt.xlabel("Ppi")
plt.ylabel("Frequency")
plt.grid(False)
```



Pixels per inch (PPI) is typically used to refer to the display resolution, or pixel density, of a computer monitor or screen. The greater the pixels per inch (PPI), the greater the detail in the image or display. Above histogram depicts ppi of laptops. Almost 690 laptops have ppi between 100 to 150. 200 laptops have ppi between 150 to 200. 40 laptops have ppi between 200 to 250. 50 laptops have ppi between 250 to 300. 20 laptops have ppi between 300 to 350

What is the distribution of laptops based on the presence or absence of In-Plane Switching ('Ips') technology, and what percentage of laptops in the dataset fall into each category?

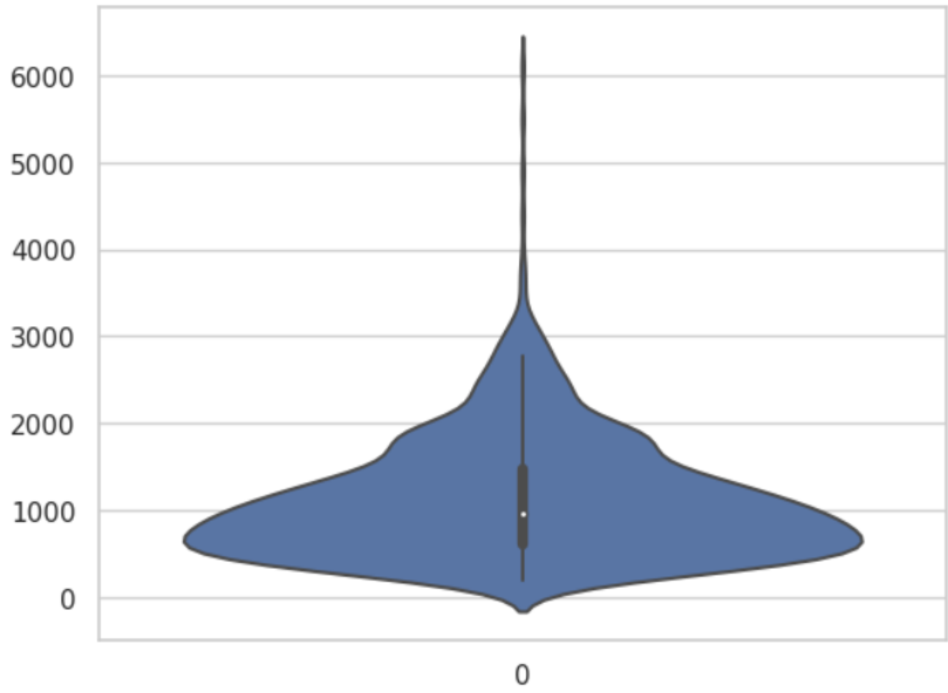
```
x = df['Ips'].value_counts()
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.show()
```



the pie chart provides a visual representation of the distribution of laptops based on the presence or absence of IPS technology. It helps viewers understand the prevalence of IPS displays among the laptops in the dataset, making it easier to draw insights about this feature's representation in the data.

What does the distribution of laptop prices look like, and are there any prominent patterns, skewness, or outliers in the dataset?

```
sns.violinplot(df['Price_euros'],orient='vertical')
```

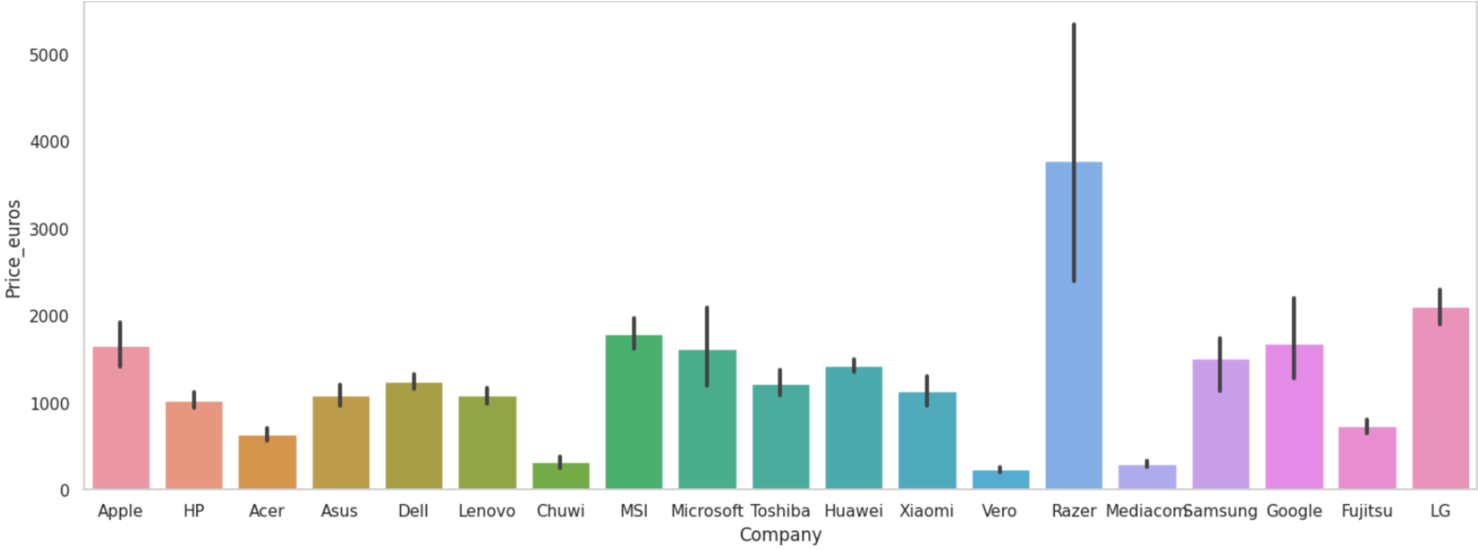


Above violin plot depicts the prices of laptops. Prices of laptops starts from 0 to above 6000. Most of laptops have prices between 0 to 2000.

Bivariate Analysis

How do laptop prices ('Price_euros') vary among different laptop manufacturers ('Company'), and are there any notable price differences or trends?

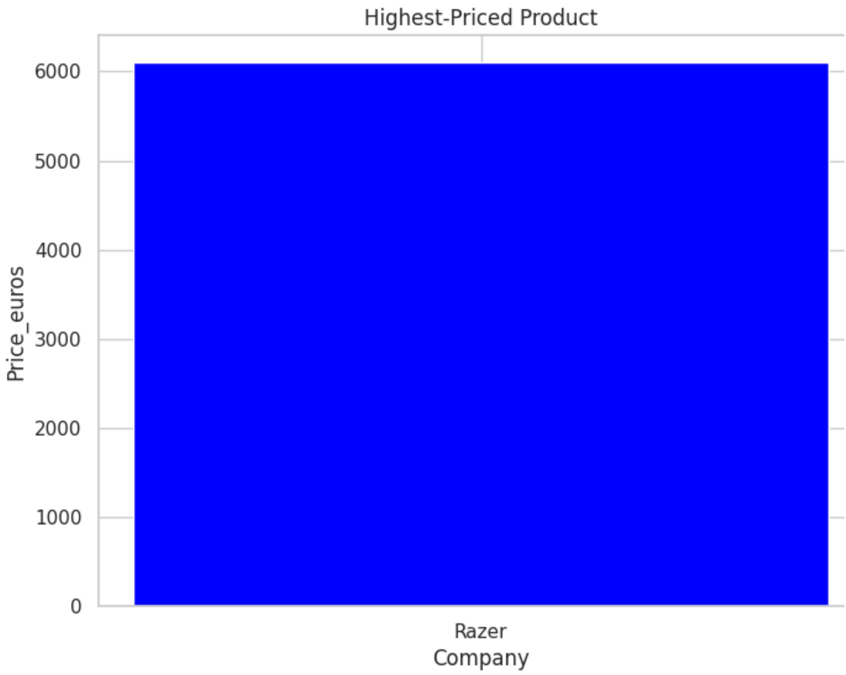
```
plt.figure(figsize=(17,6))

sns.barplot(x='Company',y='Price_euros',data=df)

plt.grid(False)

plt.show()
```



the bar plot offers a clear comparison of laptop prices across various manufacturers. It provides valuable information about the price distribution and allows for the identification of manufacturers with different pricing strategies within the laptop market.

What does the graph depicts?

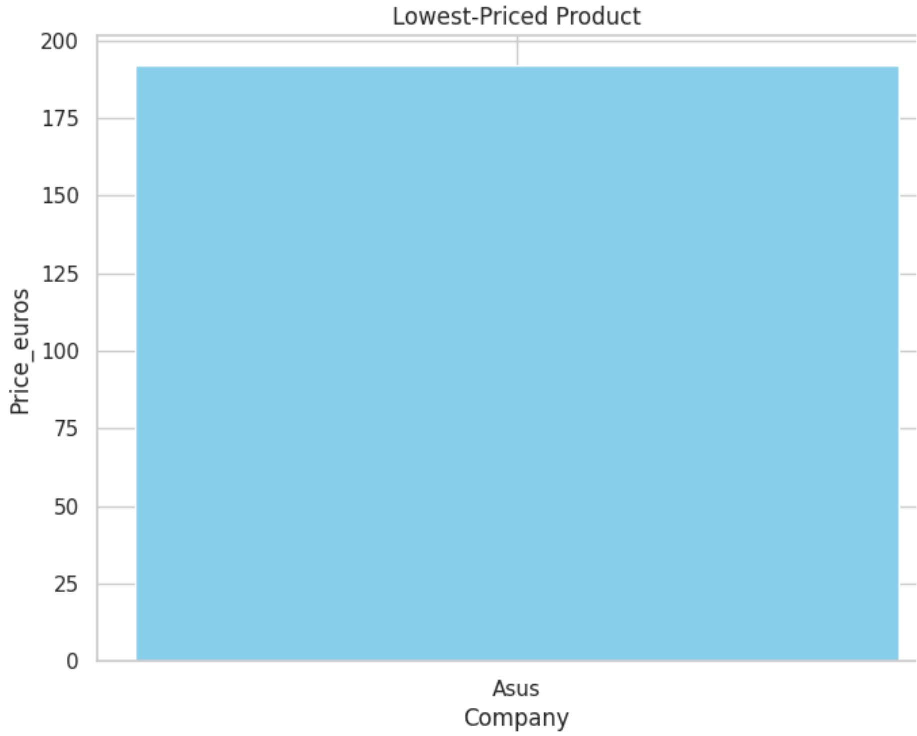
```
highest_price_product = df[df['Price_euros'] == df['Price_euros'].max()]
highest_product_name = highest_price_product['Company'].values[0]
highest_product_price = highest_price_product['Price_euros'].values[0]
plt.figure(figsize=(8, 6))
plt.bar(highest_product_name, highest_product_price, color='blue')
plt.xlabel('Company')
plt.ylabel('Price_euros')
plt.title('Highest-Priced Product')
plt.show()
```



Above graph shows the laptop which has highest price

What does the graph depicts?

```
lowest_price_product = df[df['Price_euros'] == df['Price_euros'].min()]
lowest_product_name =lowest_price_product['Company'].values[0]
lowest_product_price = lowest_price_product['Price_euros'].values[0]
plt.figure(figsize=(8, 6))
plt.bar(lowest_product_name, lowest_product_price, color='skyblue')
plt.xlabel('Company')
plt.ylabel('Price_euros')
plt.title('Lowest-Priced Product')
plt.show()
```



Above graph shows the laptop which has lowest price

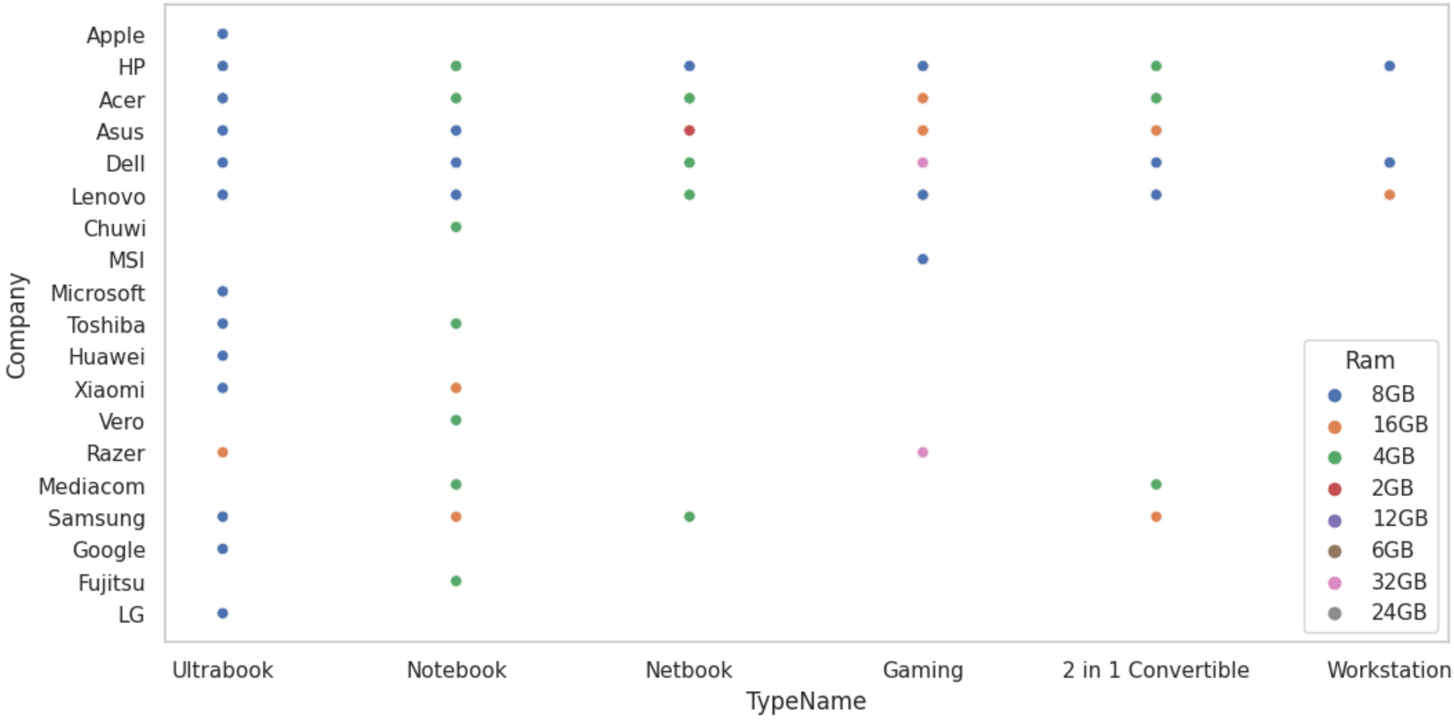
With the help of scatter plot show which laptop holds how many Gb Ram

plt.figure(figsize=(12,6))

sns.scatterplot(x='TypeName',y='Company',hue='Ram',data=df)

plt.grid(False)

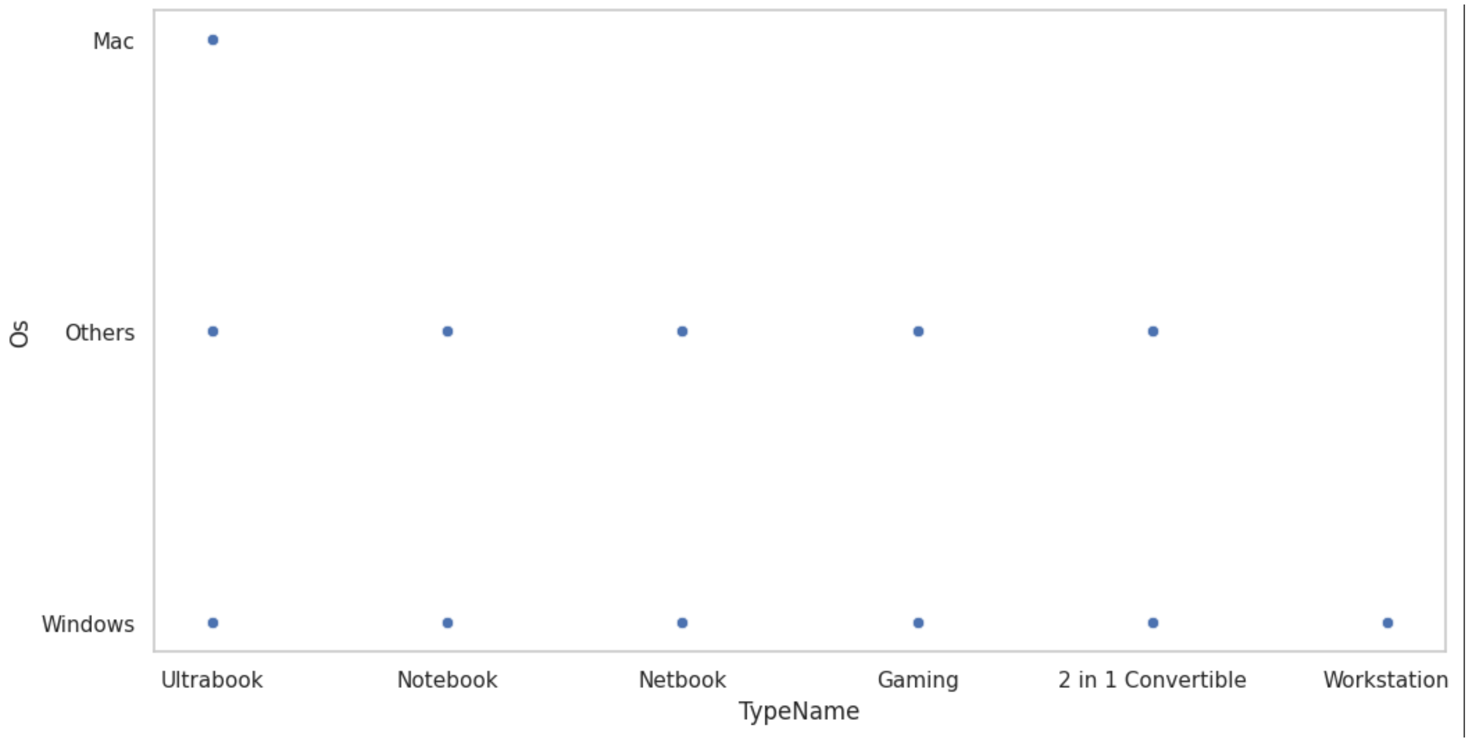
plt.show()



Scatterplot shows the Ram hold by company with its type name

How is the distribution of laptop types ('TypeName') influenced by the choice of operating system ('Os') in the dataset?

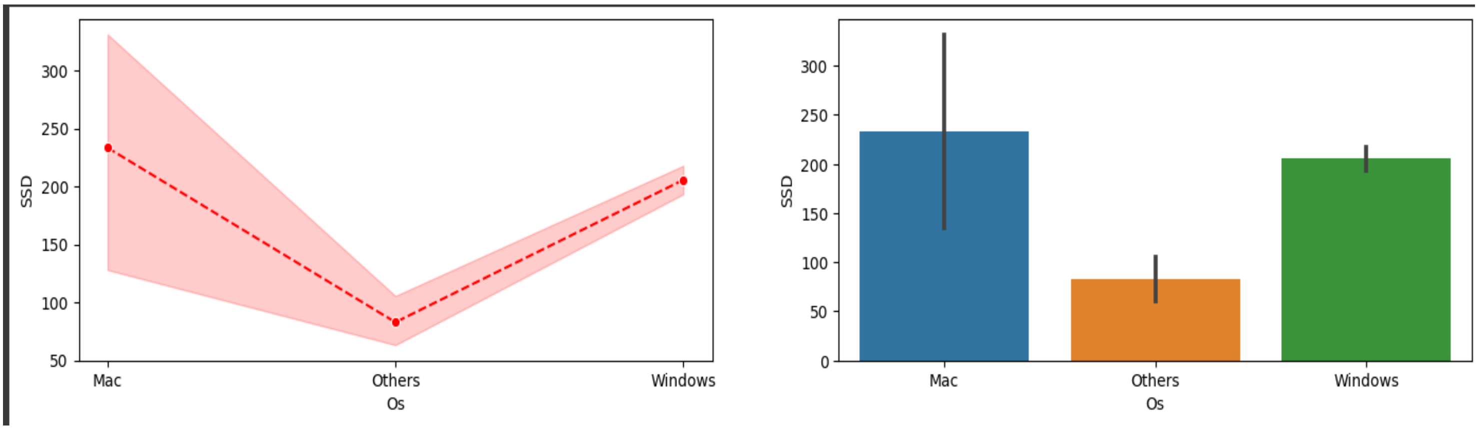
```
plt.figure(figsize=(12,6))
sns.scatterplot(x='TypeName',y='Os',data=df)
plt.grid(False)
```



The swarm plot provides an overview of how SSD storage capacities are distributed across different laptop types. Each point in the plot represents an individual laptop's SSD capacity within its respective type.

How does the choice of operating system ('Os') influence the presence of Solid-State Drives ('SSD') in laptops, and how does this influence vary between the two types of plots (line plot and bar plot)?

```
plt.figure(figsize=(17,8))
plt.subplot(2,2,1)
plt.subplot(2,2,2)
sns.barplot(x='Os',y='SSD',data=df)
plt.grid(False)
```



both plots visualize how the choice of operating system relates to the presence of SSDs in laptops. The line plot emphasizes trends and variations, while the bar plot provides a clearer comparison of SSD presence among different operating systems. These visualizations can help in assessing the impact of the operating system choice on the inclusion of SSDs in laptops.

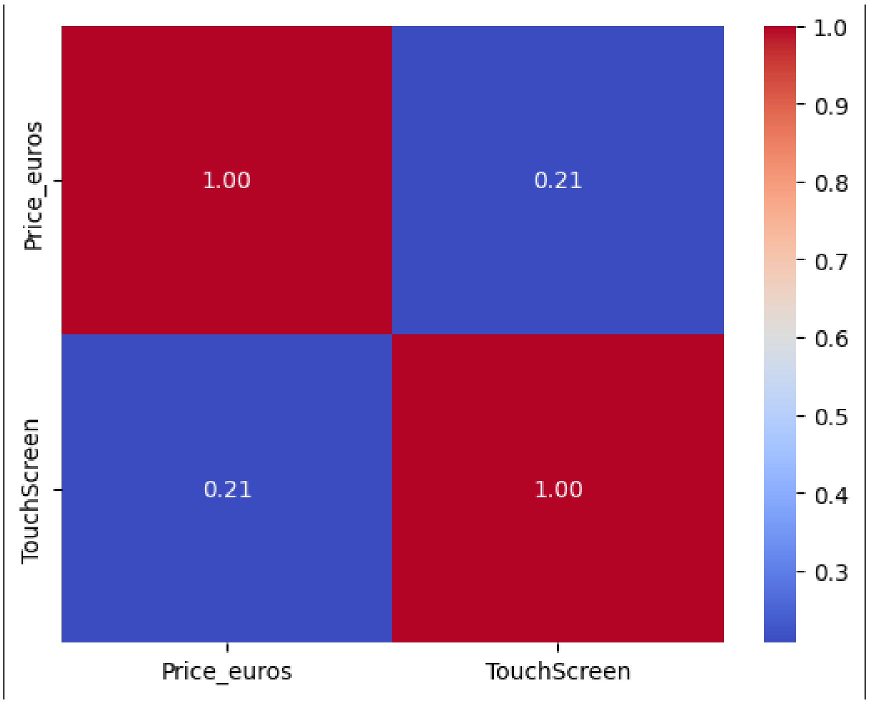
What is the correlation between 'Price_euros' and the presence of a 'TouchScreen,' and does this correlation exhibit any secondary correlations among the correlation values?

corr_matrix=df[['Price_euros','TouchScreen']].corr()

sns.heatmap(corr_matrix,annot=True,cmap='coolwarm',fmt=".2f")

print(corr_matrix.corr())

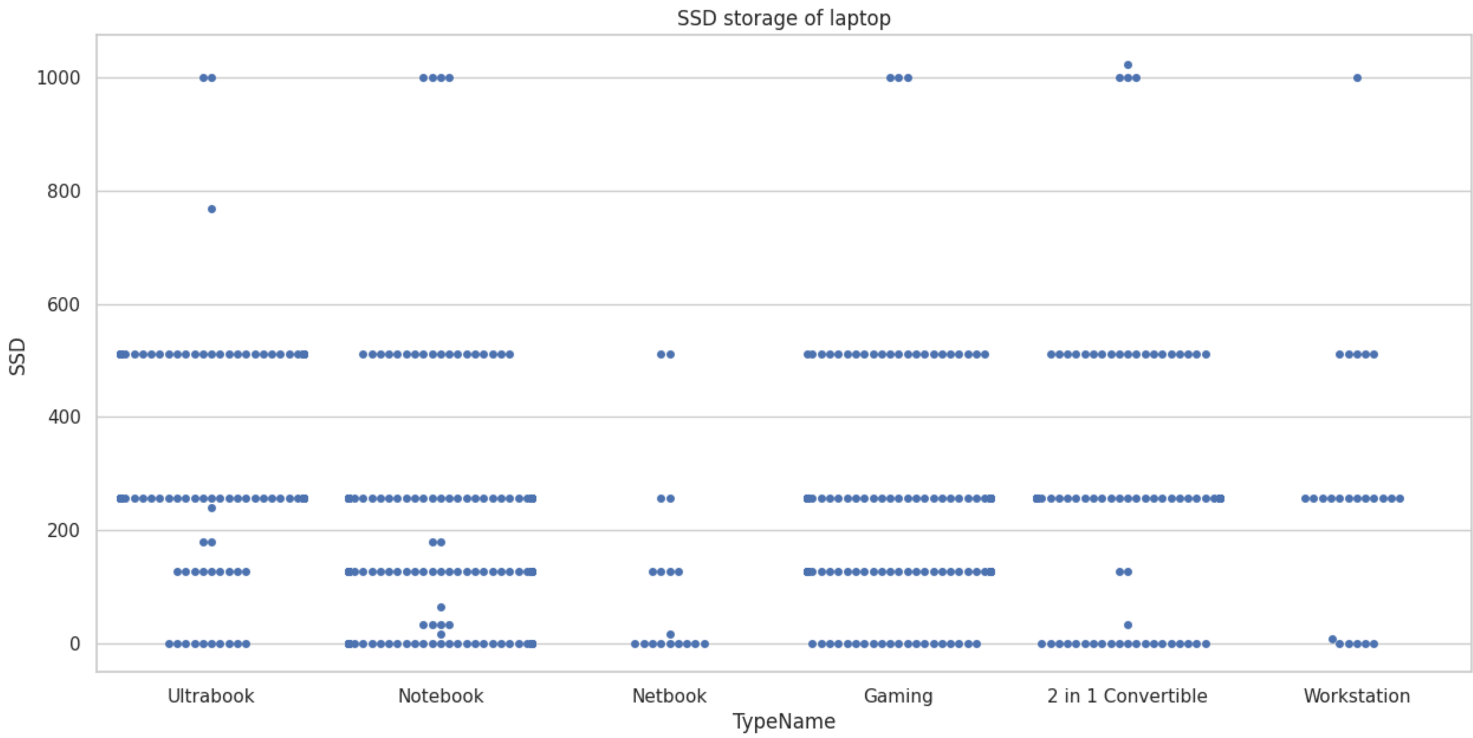
```
             Price_euros  TouchScreen
Price_euros          1.0         -1.0
TouchScreen         -1.0          1.0
```



the analysis indicates that the presence of a touchscreen in a laptop is not strongly correlated with its price in this dataset. Additionally, there is no secondary correlation observed in the meta-correlation matrix

How is the distribution of SSD storage capacity (Solid State Drive) across different laptop types (TypeName) represented in the dataset?

```
plt.figure(figsize=(15,7))
sns.swarmplot(x='TypeName',y='SSD',data=df)
plt.title('SSD storage of laptop')
```



The swarm plot provides an overview of how SSD storage capacities are distributed across different laptop types. Each point in the plot represents an individual laptop's SSD capacity within its respective type.

variance=df.var()

print("variance:")

print(variance)

```
variance:
Price_euros    497561.874038
TouchScreen         0.124099
Ips                 0.203114
Ppi              1808.962804
HDD            282773.986386
SSD             35589.306811
dtype: float64
```

cov_matrix=df.cov()

print("Covariance matrix:")

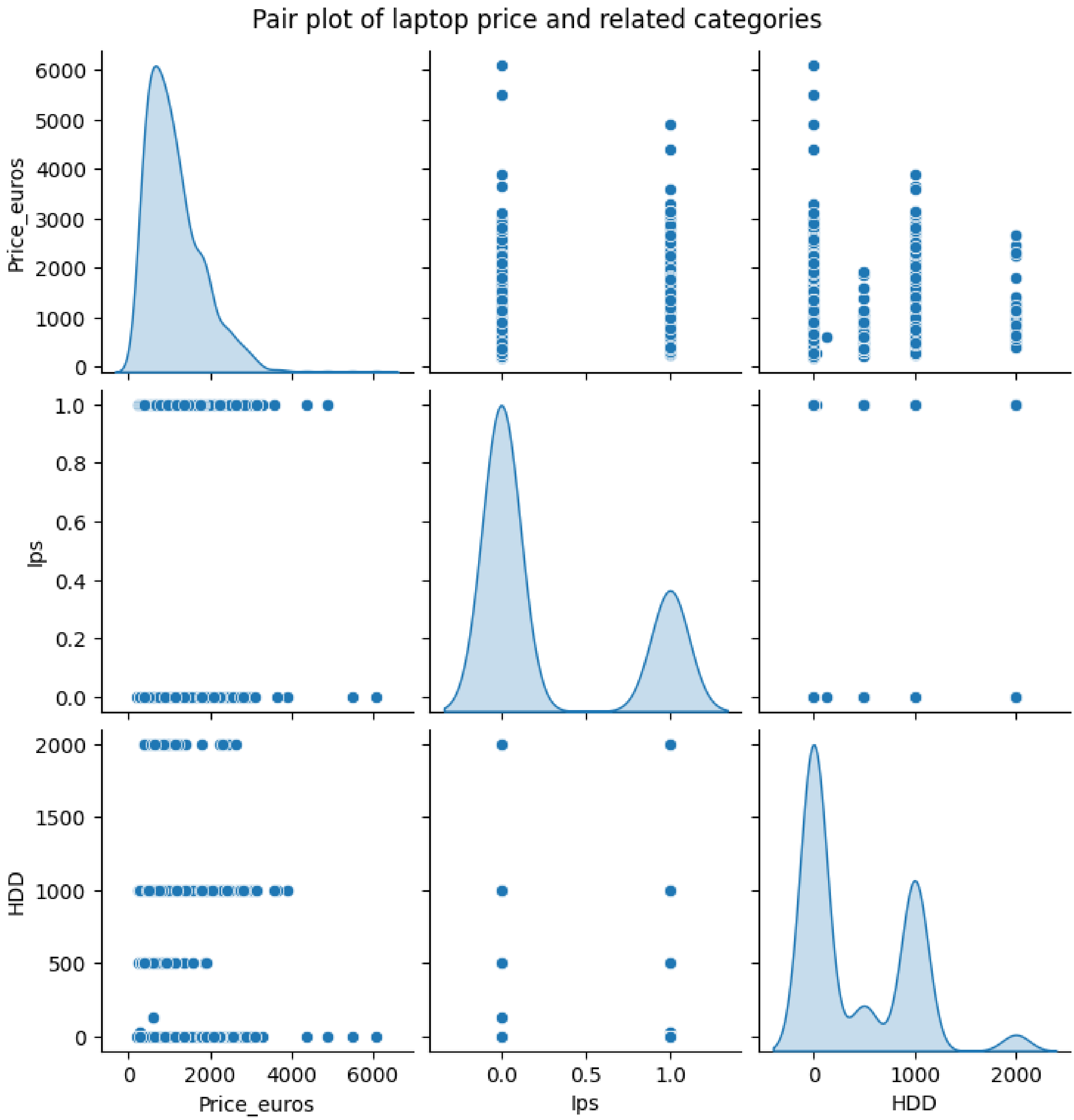
print(cov_matrix)

```
Covariance matrix:
               Price_euros  TouchScreen        Ips          Ppi  \
Price_euros  497561.874038    51.382993  74.327159  14953.386597
TouchScreen      51.382993     0.124099   0.015981      6.571629
Ips              74.327159     0.015981   0.203114      5.820017
Ppi           14953.386597     6.571629   5.820017   1808.962804
HDD          -32183.301568   -34.740440 -25.974755  -6221.209517
SSD           89458.134470    16.220941  18.235724   4204.679108

                       HDD           SSD
Price_euros  -32183.301568  89458.134470
TouchScreen     -34.740440     16.220941
Ips             -25.974755     18.235724
Ppi           -6221.209517   4204.679108
HDD          282773.986386 -39290.412653
SSD          -39290.412653  35589.306811
```

Multivariate Analysis

How do the variables 'Ips' and 'HDD' relate to the 'Price_euros' of laptops?

```
sns.pairplot(df[['Price_euros','Ips','HDD']],diag_kind='kde')
plt.suptitle("Pair plot of laptop price and related categories",y=1.02)
```



Overall, this pair plot indicates that while IPS displays may have some influence on laptop prices, HDD storage capacity does not appear to be a significant factor in determining laptop prices.

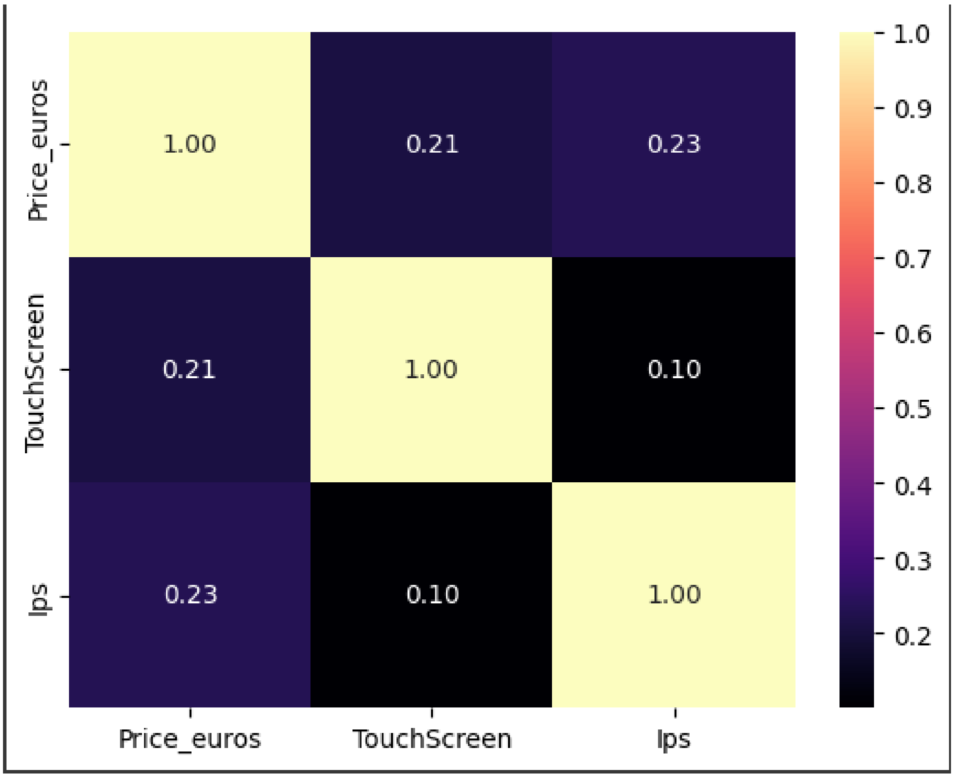
What are the correlations between 'Price_euros,' 'TouchScreen,' and 'Ips,' and do these features exhibit any secondary correlations among themselves in the context of this dataset?

corr_matrix=df[['Price_euros','TouchScreen','Ips']].corr()

sns.heatmap(corr_matrix,annot=True,cmap='magma',fmt=".2f")

print(corr_matrix.corr())

```
             Price_euros  TouchScreen       Ips
Price_euros     1.000000    -0.430846 -0.348531
TouchScreen    -0.430846     1.000000 -0.695678
Ips            -0.348531    -0.695678  1.000000
```



From,the correlations observed in this dataset, it appears that neither the presence of a touchscreen nor the presence of an IPS display has a strong linear correlation with laptop prices. Additionally, there doesn't seem to be a strong secondary correlation between 'TouchScreen' and 'Ips' or any other significant interrelationship among the variables 'Price_euros,' 'TouchScreen,' and 'Ips' in this dataset

Conclusion

In overall visualization ,we can see distribution of features of different types of laptop

Razer laptop is high priced

Asus laptop is low priced

Notebook type of laptop is sold in large number

MacOs has SSD (SSDs are a modern storage technology known for their speed, durability, and energy efficiency, making them a popular choice for both consumer and enterprise computing devices.)

Among all laptop types,ultrabook has MacOs

Link:

https://colab.research.google.com/drive/1TOVWHKXXKEE0qjT4tFi1FEETOQqZkJNm?usp=sharing

https://github.com/Preetib003/Preetib003/blob/main/data_visualization_project.ipynb